Indian Institute of Technology Patna
CS1101- Foundations of Programming

Lab6: ID array, Array Pitfalls and Programming

Date: 1-9-2025

Work in a separate directory named **Lab6**. You may use **gedit** & Editor (instead of **vi/nano**).

Task1: Carefully explore the provided *L6_array.c* program. Study each C construct demonstrated in the code, and document in your lab record any constructs or features that are new to you, along with brief notes on your observations.. (10 Points)

Task2: Carefully explore the provided program L6_arrayPitfalls.c. Examine each C construct and identify the common pitfalls or tricky behaviors illustrated in the code. In your lab record, document the constructs or features that are new or confusing to you, and include brief notes on your observations about why they may lead to errors or unexpected results.

(10 Points)

Task3: (20 Points)

Read in a number x(double), the number of terms n (int), and k (integer). Print (k+1) values of cos(x) equally spaced in the range [0,2π], considering up to n terms of the Taylor series for each point using the given expression.

For k=4, you need to print for 0π, 0.5π, 1π,1.5π,2π.

For n=5, the expression for cos(x) is as follows:

$\cos(x)=1-x^2/2!+x^4/4!-x^6/6!+x^8/8$

For example

Enter a number x (double): 0

Enter the number of terms n (int): 5

Enter the number k (int): 4

```
Evaluating cos(x) with n=5 terms at (k+1)=5 points in [0, 2π]:

x = 0.000000000000000 rad  (0.000 π)  ->  cos(x) ≈ 1.000000000000000

x = 1.570796326794897 rad  (0.500 π)  ->  cos(x) ≈ -0.000894522998474

x = 3.141592653589793 rad  (1.000 π)  ->  cos(x) ≈ -1.000003542584286

x = 4.712388980384690 rad  (1.500 π)  ->  cos(x) ≈ 0.001048182646491

x = 6.283185307179586 rad  (2.000 π)  ->  cos(x) ≈ 1.000000003529080
```

Task 4: .   (20 Points)

Write a C program that prints all prime numbers within a given range [A,B].

The program should:

- Prompt the user to enter two integers A and B.
- Ensure that A≤B, use branching to swap their values.
- Use a `for` loop to check the primality of each number in the range.
- Print all the prime numbers between A and B.

For example:

```
Enter two integers A and B (A <= B): 10 30

Prime numbers between 10 and 30 are: 11 13 17 19 23 29
```

Eg2:

```
Enter two integers A and B (A <= B): 50 100

Prime numbers between 50 and 100 are: 53 59 61 67 71 73 79 83 89 97
```

Task 5:   (10 points)

Write a C program to separate the odd and even integers from a given array into two different arrays.   The program should:

- Prompt the user to enter the size of the array.
- Accept the array elements from the user.
- Store the even numbers in one array and the odd numbers in another.
- Display the contents of the even array and the odd array separately.

For example:

```
Enter the size of the array: 8

Enter 8 elements: 5 24 5 63 34 0 25 10



Even elements: 24 34 0 10

Odd elements: 5 5 63 25
```

Task6: Write a C program to count and display the frequency of each element in an array.

The program should:

- Ask the user to enter the size of the array (with a maximum limit of 100).
- Accept the array elements from the user.
- Count how many times each unique element occurs in the array.
- Display the frequency of every element that appears in the array.

(20 Points)

```
Enter the size of the array (max 100): 5
Enter 5 elements of the array: 6 7 5 6 2

2 occurs 1 time(s)
5 occurs 1 time(s)
6 occurs 2 time(s)
7 occurs 1 time(s)
```

In Record

Task 1 and Task 2

Task 3,4 , 5 and 6: Demonstrate your work to TAs. Submit your code in single file to (roll_number.c file) to

https://u.pcloud.com/#/puplink?code=J0hXZywG6FhyRQpkzgsJ6CvyoA7dVuU3V